TRABAJO PRACTICO FINAL INTEGRADO – 2012/3
PROGRAMACION I – PROGRAMACION ORIENTADA A OBJETOS – PRACTICA PROFESIONAL

Versión 9.0, 28 de Junio de 2013

Diagrama del Sistema:

I.S.F.D. y T. Nro 189
Luján

Programacion I
Programacion Orientada a Objetos
Practica Profesional
Trabajo Final 2012
Lic. Guillermo Cherencio

Servidor PHP
Paginas php publicas integradas en sitio isft189.edu.ar
que consultan / graban archivos planos
importados / exportados de SGANS
(PRACTICA PROFESIONAL)

www.isft189.edu.ar

INTERNET

Area de red local del ISFT 189

SGANSW1 aplicacion
C utilizando interfase
CGI contra servidor
Mongoose 3.1
Utiliza libreria libfb
Aplicacion de consulta
(PROGRAMACION I)

SGANSW2 aplicacion
J2EE 6 contra servidor
Glassfish 3.1.2
(PRACTICA PROFESIO-
NAL)

Servidor Local ISFT 189
Glassfish 3.1.2
Mongoose 3.1
Firebird 2.5

Secretaría
Dirección
Alumnos
Docentes

Tablas
Views
SP
SGBD Firebird 2.5

SGANS aplicacion
client-server
J2SE + Jaybird + jdbc vs
SGBD Firebird 2.5
(PROGRAMACION ORIENTADA
A OBJETOS)

Programación I

Aplicación CGI que utiliza servidor Mongoose 3.1 mas librería libfb para comunicación con base de datos firebird 2.5. Aplicación que permite realizar consultas y operaciones que no comprometen la integridad del sistema, para acceso a través de red local.

Listado de Materias Rendidas (Grupo I)

- Tablas relacionadas: sp_tbl_mesa_ex, sp_tbl_mesa_ex_al, sp_tbl_alumno
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Nro de Legajo
- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el numero de legajo (si no existe, emito tabla html con mensaje de error); si el numero de legajo existe, continuo haciendo consulta de materias rendidas, cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.

- CONSULTAS:
SELECT COUNT(*) AS CANTIDAD FROM SP_TBL_ALUMNO WHERE LEGAJOA = 1234 (reemplazar 1234 por el legajo que tipeo el usuario)

SELECT ME.FECHA, ME.LIBRO, ME.FOLIO, ME.CODIGOM, MAT.DESCR, MA.NOTA, AL.APENOM,AL.TDOC, AL.NRODOC
FROM SP_TBL_MESAEX ME INNER JOIN
SP_TBL_MESAEX_AL MA ON ME.IDM = MA.IDM INNER JOIN
SP_TBL_ALUMNO AL ON MA.LEGAJOA = AL.LEGAJOA INNER join
SP_TBL_MATERIA MAT ON ME.CODIGOM = MAT.CODIGOM
WHERE MA.LEGAJOA = 1234 AND ME.ESTADO = 'C' AND MA.NOTA >= 4
ORDER BY MAT.CODIGOM
(reemplazar 1234 por el legajo que tipeo el usuario)

- Salida:
  - Tabla html en la parte superior los datos del alumno, en el cuerpo de la tabla las materias rendidas: fecha, libro, folio, codigo materia, nombre de materia, nota

Listado de Comisiones (Aumnos inscriptos en asignaturas) (Grupo I)

- Tablas relacionadas: sp_tbl_dmc, sp_tbl_dmca, sp_tbl_alumno
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Codigo de la materia (ej. 7106, 11106; es único por carrera,año)

CONSULTAS:
SELECT COUNT(*) FROM SP_TBL_DMC DMC
WHERE EXTRACT(YEAR FROM DMC.INICIO) = EXTRACT(YEAR FROM current_date) AND DMC.FIN IS NULL AND DMC.CODIGOM = 1234;
(reemplazar 1234 por el codigo de materia)

SELECT
MAT.CODIGOM,MAT.DESCR,C.CODIGO,C.ANIO,C.LETRA,DMC.LEGAJO,DOC.APENOM AS APENOMD,DMC.INICIO,DMCA.LEGAJOA,A.APENOM,DMCA.FECHA_INS
FROM SP_TBL_DMC DMC INNER JOIN SP_TBL_DMCA DMCA ON DMC.IDDMC = DMCA.IDDMC INNER JOIN
SP_TBL_ALUMNO A ON DMCA.LEGAJOA = A.LEGAJOA INNER JOIN
SP_TBL_COMISION C ON DMC.IDC = C.IDC INNER JOIN
SP_TBL_MATERIA MAT ON DMC.CODIGOM = MAT.CODIGOM INNER JOIN
SP_TBL_DOCENTE DOC ON DMC.LEGAJO = DOC.LEGAJO
WHERE EXTRACT(YEAR FROM DMC.INICIO) = EXTRACT(YEAR FROM current_date) AND
DMC.FIN IS NULL AND DMC.CODIGOM = 1234
ORDER BY DMC.CODIGOM, A.APENOM;
(reemplazar 1234 por el codigo de materia)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el numero de materia (si no existe, emito tabla html con mensaje de error); si el numero de materia existe, continuo haciendo consulta de alumnos inscriptos en esa materia (sp_tbl_dmc con fecha de fin = null), cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - Tabla html en la parte superior los datos de la materia (codigo, descripcion), en el cuerpo de la tabla los alumnos inscriptos: fecha inscripcion, legajo, apellido, nombre.

Listado de Situación Académica de Alumno (resultado de cursadas) (Grupo I)

- Tablas relacionadas: sp_tbl_dmc, sp_tbl_dmca, sp_tbl_alumno
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Nro de Legajo

CONSULTAS:

SELECT COUNT(*) AS CANTIDAD
FROM SP_TBL_DMC DMC INNER JOIN SP_TBL_DMCA DMCA ON DMC.IDDMC = DMCA.IDDMC INNER JOIN
SP_TBL_ALUMNO A ON DMCA.LEGAJOA = A.LEGAJOA INNER JOIN
SP_TBL_MATERIA MAT ON DMC.CODIGOM = MAT.CODIGOM INNER JOIN
SP_TBL_DOCENTE DOC ON DMC.LEGAJO = DOC.LEGAJO

WHERE EXTRACT(YEAR FROM DMC.INICIO) = EXTRACT(YEAR FROM current_date) AND
DMC.FIN IS NOT NULL AND A.LEGAJOA = 1234;
(reemplazar 1234 por el codigo de alumno)

SELECT MAT.CODIGOM,MAT.DESCR,DMC.LEGAJO,DOC.APENOM AS APENOMD,DMC.INICIO,DMC.FIN,DMCA.LEGAJOA,A.APENOM,DMCA.FECHA_INS,DMCA.NOTA,DMCA.SITUACION,DMCA.OBS
FROM SP_TBL_DMC DMC INNER JOIN SP_TBL_DMCA DMCA ON DMC.IDDMC = DMCA.IDDMC INNER JOIN
SP_TBL_ALUMNO A ON DMCA.LEGAJOA = A.LEGAJOA INNER JOIN
SP_TBL_MATERIA MAT ON DMC.CODIGOM = MAT.CODIGOM INNER JOIN
SP_TBL_DOCENTE DOC ON DMC.LEGAJO = DOC.LEGAJO
WHERE EXTRACT(YEAR FROM DMC.INICIO) = EXTRACT(YEAR FROM current_date) AND
DMC.FIN IS NOT NULL AND A.LEGAJOA = 1234
ORDER BY DMC.CODIGOM;
(reemplazar 1234 por el codigo de alumno)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el numero de legajo (si no existe, emito tabla html con mensaje de error); si el numero de legajo existe, continuo haciendo consulta de alumnos inscriptos en asignaturas (sp_tbl_dmc con fecha de inicio mas tardia), cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - Tabla html en la parte superior los datos del alumno (legajo, apellido,nombre), en el cuerpo de la tabla las materias cursadas por el alumno: fecha inscripcion, codigo materia, nombre materia, nota, observaciones

Listado de Mesas de Examen por Carrera (Grupo II, hecho!)

- Tablas relacionadas: sp_tbl_mesaex, sp_tbl_materia, sp_tbl_plane, sp_tbl_docente
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Codigo de la Carrera (ej. codigo 7 “Tec. Sup en Analsis de Sistemas”)

CONSULTAS:
SELECT COUNT(*) FROM SP_TBL_PLANE P
WHERE P.CODIGO = 1234;
(reemplazar 1234 por el codigo de carrera)

```
SELECT ME.IDM, ME.FECHA, ME.ESTADO, ME.CODIGOM, MA.DESCR,
ME.TITULAR, TIT.APENOM,
        ME.VOCAL1, VOC1.APENOM, ME.VOCAL2, VOC2.APENOM, PL.CODIGO,
PL.DESCR, PL.NROR, PL.TITULO
FROM SP_TBL_MESAEX  ME
INNER JOIN SP_TBL_MATERIA MA    ON ME.CODIGOM = MA.CODIGOM
INNER JOIN SP_TBL_PLANE PL      ON MA.CODIGO = PL.CODIGO
INNER JOIN SP_TBL_DOCENTE TIT   ON ME.TITULAR = TIT.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC1  ON ME.VOCAL1 = VOC1.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC2  ON ME.VOCAL2 = VOC2.LEGAJO
WHERE ME.ESTADO IN('P','A') AND PL.CODIGO = 1234
ORDER BY ME.FECHA ASC;
```

(reemplazar 1234 por el codigo de carrera)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el numero de carrera (si no existe, emito tabla html con mensaje de error); si el numero de carrera existe, continuo haciendo consulta de mesas de examen en estado "P" o "A" (programadas o abiertas) , cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - **Tabla html en la parte superior los datos de la carrera (codigo, descripcion), en el cuerpo de la tabla las mesas programadas o abiertas de las distintas asignaturas de la carrera en cuestión: fecha, codigo de materia, descripcion de materia, profesor titular, primer vocal, segundo vocal.**

Listado de Mesas de Examen por Materia (Grupo II, hecho!)

- Tablas relacionadas: sp_tbl_mesaex, sp_tbl_materia, sp_tbl_plane, sp_tbl_docente
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Codigo de la Materia (ej. codigo 8104 "Practica Profesional I")

CONSULTAS:

```
SELECT COUNT(*) FROM SP_TBL_MATERIA P
WHERE P.CODIGOM = 1234;
```

(reemplazar 1234 por el codigo de materia)

```
SELECT ME.IDM, ME.FECHA, ME.ESTADO, ME.CODIGOM, MA.DESCR,
ME.TITULAR, TIT.APENOM,
```

ME.VOCAL1, VOC1.APENOM, ME.VOCAL2, VOC2.APENOM, PL.CODIGO, PL.DESCR, PL.NROR, PL.TITULO
FROM SP_TBL_MESAEX ME
INNER JOIN SP_TBL_MATERIA MA ON ME.CODIGOM = MA.CODIGOM
INNER JOIN SP_TBL_PLANE PL ON MA.CODIGO = PL.CODIGO
INNER JOIN SP_TBL_DOCENTE TIT ON ME.TITULAR = TIT.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC1 ON ME.VOCAL1 = VOC1.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC2 ON ME.VOCAL2 = VOC2.LEGAJO
WHERE ME.ESTADO IN('P','A') AND MA.CODIGOM = 1234
ORDER BY ME.FECHA ASC;
(reemplazar 1234 por el codigo de materia)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el numero de materia (si no existe, emito tabla html con mensaje de error); si el numero de materia existe, continuo haciendo consulta de mesas de examen en estado “P” o “A” (programadas o abiertas) , cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - **Tabla html en la parte superior los datos de la materia (codigo, descripcion, codigo y descripcion tecnicatura a la que pertenece la materia), en el cuerpo de la tabla las mesas programadas o abiertas de la materia en cuestión: fecha, profesor titular, primer vocal, segundo vocal.**

Listado de Mesas de Examen entre Fechas (Grupo III)

- Tablas relacionadas: sp_tbl_mesaex, sp_tbl_materia, sp_tbl_plane, sp_tbl_docente
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Fecha Desde (ej. “01-DEC-12” dia 01 de diciembre de 2012)
  - Fecha Hasta (ej. “26-DEC-12” dia 26 de diciembre de 2012)

CONSULTAS:

SELECT ME.IDM, ME.FECHA, ME.ESTADO, ME.CODIGOM, MA.DESCR, ME.TITULAR, TIT.APENOM,
ME.VOCAL1, VOC1.APENOM, ME.VOCAL2, VOC2.APENOM, PL.CODIGO, PL.DESCR, PL.NROR, PL.TITULO
FROM SP_TBL_MESAEX ME
INNER JOIN SP_TBL_MATERIA MA ON ME.CODIGOM = MA.CODIGOM
INNER JOIN SP_TBL_PLANE PL ON MA.CODIGO = PL.CODIGO
INNER JOIN SP_TBL_DOCENTE TIT ON ME.TITULAR = TIT.LEGAJO

```
INNER JOIN SP_TBL_DOCENTE VOC1  ON ME.VOCAL1 = VOC1.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC2  ON ME.VOCAL2 = VOC2.LEGAJO
WHERE ME.ESTADO IN('P','A') AND ME.FECHA BETWEEN FDESDE AND FHASTA
ORDER BY ME.FECHA ASC, MA.DESCR ASC;
```

(reemplazar FDESDE por la fecha Desde entre comillas simples y FHASTA por la fecha Hasta entre comillas simples)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hago consulta de mesas de examen en estado “P” o “A” (programadas o abiertas) entre las fechas desde y hasta indicadas, cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - **Tabla html en la parte superior se indica titulo, indicando las fechas desde y hasta solicitadas, en el cuerpo de la tabla las mesas programadas o abiertas entre fechas, indicando: fecha, codigo carrera, descripcion carrera, codigo materia, descripcion materia, profesor titular.**

Listado de Inscripción a Mesas de Examen (por alumno) (Grupo III)

- Tablas relacionadas: sp_tbl_mesaex, sp_tbl_materia, sp_tbl_plane, sp_tbl_docente, sp_tbl_alumno, sp_tbl_mesaex_al
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Legajo de Alumno (ej. legajo 5 “Gonzalez, Anabella”)

CONSULTAS:

```
SELECT COUNT(*) FROM SP_TBL_ALUMNO A
WHERE A.LEGAJOA = 1234;
```

(reemplazar 1234 por el legajo del alumno)

```
SELECT ME.IDM, ME.FECHA, ME.ESTADO, ME.CODIGOM, MA.DESCR, ME.TITULAR, TIT.APENOM,
       ME.VOCAL1, VOC1.APENOM, ME.VOCAL2, VOC2.APENOM, PL.CODIGO, PL.DESCR, PL.NROR,
           PL.TITULO, AL.LEGAJOA, AL.APENOM, AL.TDOC, AL.NRODOC, MAL.FECHA_INS
FROM SP_TBL_MESAEX  ME
INNER JOIN SP_TBL_MATERIA MA    ON ME.CODIGOM = MA.CODIGOM
INNER JOIN SP_TBL_PLANE PL      ON MA.CODIGO  = PL.CODIGO
INNER JOIN SP_TBL_DOCENTE TIT   ON ME.TITULAR = TIT.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC1  ON ME.VOCAL1  = VOC1.LEGAJO
INNER JOIN SP_TBL_DOCENTE VOC2  ON ME.VOCAL2  = VOC2.LEGAJO
```

```
INNER JOIN SP_TBL_MESAEX_AL MAL ON ME.IDM     = MAL.IDM
INNER JOIN SP_TBL_ALUMNO AL     ON MAL.LEGAJOA = AL.LEGAJOA
WHERE ME.ESTADO IN('P','A') AND AL.LEGAJOA = 1234
ORDER BY ME.FECHA ASC,MA.DESCR ASC;
```

(reemplazar 1234 por el legajo del alumno)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el legajo del alumno (si no existe, emito tabla html con mensaje de error); si el alumno existe, continuo haciendo consulta de mesas de examen en estado “P” o “A” (programadas o abiertas) , cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - **Tabla html en la parte superior los datos del alumno (legajo, apellido nombre), en el cuerpo de la tabla las mesas programadas o abiertas del alumno en cuestión: fecha de la mesa de examen, codigo materia, descripcion materia, profesor titular, primer vocal, segundo vocal,fecha inscripcion del alumno.**

Listado de Planillas de Asistencia por Asignatura (Grupo III)

- Tablas relacionadas: sp_tbl_materia, sp_tbl_plane, sp_tbl_docente, sp_tbl_alumno, sp_tbl_dmc, sp_tbl_dmca
- Entrada:
  - Datos de conexión a firebird (server,b.d.,usuario,contraseña,rol) (cgi3.c)
  - Codigo de Asignatura (ej. codigo 11107 “Sistemas de Computacion”)

CONSULTAS:

```
SELECT COUNT(*) FROM SP_TBL_MATERIA P
WHERE P.CODIGOM = 1234;
```

(reemplazar 1234 por el codigo de materia)

```
SELECT ASI.IDDMC, ASI.IDC, ASI.LEGAJO, D.APENOM, ASI.CODIGOM,
M.DESCR, M.ANIO, M.TIPOMAT, M.PERIODO, P.CODIGO, P.DESCR, P.NROR,
P.TITULO,
ASI.INICIO,A.LEGAJOA,A.APENOM,A.TDOC,A.NRODOC
FROM SP_TBL_DMC  ASI
INNER JOIN SP_TBL_DMCA ASIA ON ASI.IDDMC = ASIA.IDDMC
INNER JOIN SP_TBL_DOCENTE D ON ASI.LEGAJO = D.LEGAJO
INNER JOIN SP_TBL_MATERIA M ON ASI.CODIGOM = M.CODIGOM
INNER JOIN SP_TBL_PLANE   P ON M.CODIGO = P.CODIGO
INNER JOIN SP_TBL_ALUMNO  A ON ASIA.LEGAJOA = A.LEGAJOA
```

WHERE EXTRACT(YEAR FROM ASI.INICIO) = EXTRACT(YEAR FROM CURRENT_DATE) AND
ASI.FIN IS NULL AND M.CODIGOM = 1234
ORDER BY A.APENOM ASC;
(reemplazar 1234 por el codigo de materia)

- Proceso:
  - Conectar a b.d. Firebird usando los datos de conexión, hacemos consulta a b.d. Chequeando el codigo de la materia (si no existe, emito tabla html con mensaje de error); si la materia existe, continuo haciendo consulta de asignaturas que tengan fecha de inicio de este año y no tengan fecha de finalización (asignatura abierta, con alumnos aun no calificados), cargo en memoria el resultado de la consulta (fbquery1.c) y armo tabla html con los datos cargados en memoria. Termino de armar la tabla, libero memoria e imprimo todo el codigo html de la pagina.
- Salida:
  - **Tabla html en la parte superior los datos de la asignatura (codigo de carrera a la que pertenece la asignatura, descripcion de carrera, año de la carrera, codigo materia, descripcion materia, tipo, periodo,fecha inicio), en el cuerpo de la tabla los alumnos inscriptos en la asignatura en cuestión y una serie de casilleros para que el docente ponga (una vez impreso este reporte) "P" de presente o "A" de ausente. En la parte superior hacer también una grilla para que el docente pueda poner la fecha. Datos a emitir: legajo del alumno, Apellido – Nombre del alumno.**

(Grupo I)=Del Conte, Feijoo, Honeker
(Grupo II)=Schwindt (2)
(Grupo III)=??

Programación Orientada a Objetos

Aplicación J2SE swing que utiliza driver jdbc jaybird para comunicación con base de datos firebird 2.5 (isft.fdb) en servidor aplicativo, tiene todas las operaciones fundamentales para la gestión académica. La aplicación tendrá las siguientes opciones:

Menu Alumnos:

Abm Alumnos (tbl_alumnos) (Grupo I, Gomez)
Abm Mesas de Examen (tbl_mesaex) (Grupo I, Gomez)
Inscripción a Mesas de Examen (tbl_mesaex_al) (Grupo I, Gomez)
Inscripción a Asignaturas (tbl_dmca)(Grupo I, Mosca)
Copiar Inscripción de una Asignatura a Otra (tbl_dmca) (Grupo I, Mosca)
Cambios de Carrera
Comunicar a Alumnos (flexible selección de alumnos y envío de mails via interfase isft189.edu.ar, practica profesional)

Menu Docentes
Abm Docentes (tbl_docentes) (Grupo I, Gomez)
Actualizar Titulos Docentes (tbl_docentet) (Grupo I, Gomez)
Abm Asignatura (tbl_dmc)
Copiar Asignatura de año anterior (tbl_dmc)
Comunicar a Docentes (flexible selección de docentes y envío de mails via interfase isft189.edu.ar, practica profesional)

Menu Carreras:
Abm Carreras (tbl_plane, tbl_pea) (Grupo I, Gomez)
Abm Comisiones por Carrera (tbl_comision)
Abm Materias por Carrera (tbl_materia) (Grupo I, Mosca)
Correlatividades (tbl_mc) (Grupo I, Mosca)

Menu Herramientas:
Publicar Mesas de Examen (via ftp a servidor isft189.edu.ar, practica profesional)
Publicar Programas Asignaturas / Material Asignaturas (via ftp a servidor isft189.edu.ar, practica profesional)
Exportar Datos (practica profesional)
Importar Datos (practica profesional)
Inicializar Sistema (pide clave numerica y ejecuta stored procedure SP_INIT, practica profesional)

Menu Ayuda:
Manual Técnico del Sistema (instalación, desinstalación, cambios de claves firebird, backup firebird, restore firebird, datos iniciales del sistema)
Ayuda para el usuario (uso general de opciones e interfases del sistema)

(Grupo I)=Mosca, Gomez
(Grupo II)=

Practica Profesional

*Desarrollo de interfase en aplicación J2SE swing utilizando FTP para la exportación de archivos desde base de datos firebird 2.5 a servidor web www.isft189.edu.ar para luego ser utilizado por paginas php. Exportar archivo de mesas de examen: (carrera,materia,idcarrera,idmateria,fecha,legajo,apenom,titular,vocal1,vocal2,ltitutar,lvocal1, lvocal2) . Este archivo sera de texto, delimitado. Verificar los permisos de este archivo, no debería ser accesible por el publico en general, solo por quien lo creo y las paginas php que lo utilizaran. Carrera y materia contienen los nombres de carrera. Idcarrera y idmateria contienen los códigos de carrera y materia. Legajo sea 0 cuando no haya alumnos inscriptos, apenom sera NULL cuando no haya alumnos inscriptos. Titular, vocal1, vocal2 contiene los apellidos

– nombre de los docentes que están en la mesa como titular, primer y segundo vocal. Ltitular, lvocal1,lvocal2 contiene los números de legajos correspondientes de los docentes.

*Desarrollo de stored procedures en base de datos SGANS isft189.fdb para abm de tablas (en base a stored procedure de ejemplo en b.d. Test.fdb, tabla TEST):

- TBL_PEA y TBL_PLANE (pedir también la cantidad de años de la carrera al dar de alta, baja, cambio)
- TBL_MESAEX (mesas de examen)
- TBL_MESAEX_AL (mesas de examen alumno, altas bajas de inscripción a mesa de examen por un lado y por otro, otro stored procedure para poner la nota)
- TBL_MC (correlatividades)
- TBL_MATERIA
- TBL_DOCENTE
- TBL_DOCENTET (solo altas y bajas de títulos)
- TBL_DMCA (solo altas y bajas)
- TBL_DMC (solo altas y bajas)
- TBL_COMISION
- TBL_ALUMNO
- P_INIT (Recibe como argumento un valor entero, si dicho valor coincide con un determinado valor arbitrario, por ejemplo, “981981”, continua su ejecucion, sino termina. Este stored procedure inicializa la base de datos con datos de prueba. Ejecuta P_DELETE y luego ejecuta P_INSERT). Solo puede ser ejecutado por usuario sysdba.
- P_DELETE (borra todos los datos del sistema, requiere (igual que P_INIT) se le pase un argumento entero clave para ejecutar el borrado de datos, sino no hace nada). Solo puede ser ejecutado por usuario sysdba.
- P_INSERT (no requiere argumento clave, inserta datos de prueba en el sistema para comenzar a trabajar con el mismo. Esto debe ser documentado en el manual del sistema). Solo puede ser ejecutado por usuario sysdba.

*Desarrollo de paginas php de consulta para ser utilizadas desde la web en servidor www.isft189.edu.ar a partir de archivos previamente exportados:

- Consulta por materia, utilizando las primeras letras del nombre de la materia (filtra mesas de examen para una materia en particular o bien, si no se indica materia, trae todas las materias, ordenadas por carrera y materia).
- Consulta por carrera (filtra mesas de examen de una carrera en particular, debe ingresar parte del nombre de la carrera)
- Consulta por Docente, utilizando las primeras letras del apellido - nombre del docente, busca su ocurrencia ya sea como titular, vocal1 o vocal2.
- Consulta por fecha (muestra todas las mesas a una fecha determinada o bien todas ordenadas por fecha en forma ascendente)

En todos los casos, mostrar los casos mostrar el nombre de la carrera, nombre de la materia y su código, nombre del alumno (no mostrar su legajo) si hubiera inscriptos, fecha, apellido nombre de los docentes (no mostrar su legajo)

*Desarrollo J2EE 6. Tecnologías propuestas: html, javascript, ajax, jquery, jquery-ui, jtable, json, servlets, jpa 2.0 o jdbc, driver jdbc jaybird; uso de servidores glassfish o jetty embebido. Desarrollo de los siguientes módulos:

* Modulo intranet de registro de aspirantes. Permitirá almacenar (por año) los aspirantes a ingreso, datos de contacto: apellido, nombre, e-mail, TE, CEL, dirección, etc. (la mayoría serán datos opcionales). Permite borrar todos los aspirantes de un año determinado, con previa confirmación, ya sea de una carrera o todas las carreras. Permite ingresar, borrar, modificar aspirantes de un año determinado a una carrera determinada. Permite mover un aspirante de una carrera a otra. Permite listar, imprimir o genera pdf con aspirantes de un año determinado, de una carrera determinada o de todas las carreras. (German Mosca)

* Modulo intranet de inscripción a mesas de examen. El modulo asume que se trata de un alumno ejecutando esta aplicación desde la intranet del Instituto. Solicita su numero de legajo, permite actualizar solo sus datos elementales (no permite cambio de carrera, por ejemplo), muestra las mesas de examen habilitadas por materia y fecha (abiertas, con las menos 48hs de anticipación), acorde con las asignaturas que ha regularizado y aun no ha rendido final, permite que seleccione una o varias y presione el boton inscripción. El sistema emite un comprobante (pdf) de inscripción, el cual podrá ser impreso o descargado en su maquina para su resguardo. El modulo también permite “des-inscribirse” de una mesa de examen (48 hs anticipación, no emite comprobante). (Emmanuel Cassan)

* Modulo intranet de administración de mesas de examen. El modulo asume que se trata de un docente o un directivo del Instituto. Solicita tipo documento, numero y contraseña (esto debe coincidir con la tabla tbl_docente). Permite el cambio de contraseña. Si se trata de un docente, el sistema muestra la lista de mesas de examen en las cuales es titular y se encuentran abiertas, permite su selección (permite indicar el libro y folio asociado con la mesa) y luego muestra el listado de alumnos de dicha mesa para cargar la calificación de cada alumno (permite modificar cuantas veces quiera las calificaciones). En el caso de un directivo, el sistema mostrará la lista de mesas de examen por carrera que se encuentran abiertas, podrá consultar los alumnos evaluados en cada una (solo ver el listado y sus calificaciones, si las tuvieran cargadas, no permite cambios) y permitirá cerrar la mesa de examen (solicita confirmación). También permitirá abrir un nueva mesa de examen y podrá imprimir el acta volante (pdf,listado de alumnos inscriptos en mesa de examen abierta) y acta definitiva (pdf,listado de alumnos evaluados en mesa de examen cerrada, indicando libro, folio, alumnos, notas, docentes titulares y vocales). (David Gomez)

* Modulo intranet de administración de comisiones. El modulo asume que se trata de un docente o un directivo del Instituto. Si es un directivo, solicita tipo documento, numero y contraseña (esto debe coincidir con la tabla tbl_docente). Permite el cambio de contraseña. Selecciona Carrera, trae todas las comisiones (tbl_comision) de dicha carrera (para todos sus años), selecciona comisión, selecciona docente, selecciona materia (solo muestra las de la carrera y las del año de la comisión), ingresa la fecha de inicio (no posee

fecha de finalización) y permite dar de alta la comisión. Un proceso similar se sigue para seleccionar las comisiones que están iniciadas y aun no finalizadas, permite ver los alumnos asociados a cada comisión y su condición (no permite cambiar estos datos), solicita fecha de finalización, finaliza la comisión (da por terminada la cursada, para luego poder crear mesas de examen final para dicha asignatura) y emite el acta de fin de cursada (pdf). El directivo también puede agregar o quitar alumnos a una comisión iniciada. En caso de ser docente, se comienza igual que un directivo, pero éste no puede crear la comisión (poniendo fecha de inicio) ni terminarla, solo puede asociar alumnos a su comisión (alumnos que sean de la carrera en cuestión y que no tengan aprobada ni regularizada esta asignatura) y cargar sus calificaciones y situación académica final (digamos que, al final del año el docente deja todo listo para que el directivo cierre la cursada). (Santiago De La Cruz)

Atte. Lic. Guillermo Cherencio
Programción I
Programación Orientada a Objetos
Práctica Profesional
ISFT 189 www.isft189.edu.ar
www.grch.com.ar